Subject ______ "المحاضرة الرابعة" Date ______

تصحيح الأمر Stat: المراد تكراره يمكن أن يكون الأمر مفرد أو يمكن أن يكون الأمر مركب، أي في هذه الحالة نضع الأمر بين قوسين البداية والنهاية.

تمرين: أكتب برنامج يطبع الأعداد الصحيحة من 1 إلى 10 ؟

```
# include <iostream.h>
# define n 10
void main( )
{
  int i;
  for( i=1 ; i<=n ; ++i)
   cout << i;
}
```

(← ; : نهاية الأمر ونهاية حلقة for)

- حيث البداية ينفذ الوسيط الأول لإعطاء قيمة ابتدائية ثم بعد ذلك ينتقل التنفيذ الى الوسيط الثالث من أجل زيادة العداد وبعد ذلك ينفذ الوسيط الثاني من أجل التنفيذ الشرط فإذا كان الشرط محقق ينفذ أمر الطباعة.

يمكن أن تكون الحلقة متناقصة أي يتم طباعة الأعداد الصحيحة من العشرة (10 ← 1) وصيغتها:

(i!=0) ، i≠0; أو i>1

```
for(i=n ; i>0 ; --i)
    cout <<i;
```

تمرين: أكتب برنامج يحسب مجموع الأعداد من 1 إلى 10 ؟

```
#include <iostream.h>
# define n 10
void main( )
{
   int i,s; s=0;
   for (i=1 , i<=n ; ++i)
       s=s+i;
   cout<<"\n s="<<s;
}
```

كل متغير ليس له قيمة قيمته صفر

Subject .... Date ....

التنفيذ:

$i=1 \rightarrow S=0+1=1$

$i=2 \rightarrow S=1+2=3$

$i=3 \rightarrow S=3+3=6$

$i=4 \rightarrow S=6+4=10$

تمرين: أكتب برنامج يحسب $n!$:

```
#include <iostream.h>
void main()
int n, i, f;      // n!
f=1;
cout<<"\n n="; cin>>n;
for(i=1; i<=n; ++i)
f=f*i;
cout<<"\n f="<<f;
```

* حلقة While:

وصيغتها:

```
While (exp)
stat;
```

exp: هو تعبير منطقي نتيجته صح أو خطأ، فإذا كان التعبير المنطقي (exp) صحيحًا ينفذ الأمر stat، وعند عدم تحقق الشرط (exp) الأمر stat لم ينفذ.

تمرين: طباعة الأعداد 1,2,3,4 باستخدام حلقة While:

```
#include <iostream.h>
void main ( )
{
int i,n; i=0; S=0;
cin >> n;
While (i<n)
{ i=i+1;  S=S+i;
cout<<i;
}
cout<<S;
```

Subject Date

تمرين: أكتب برنامج يسمح بإدخال عدة أعداد صحيحة من لوحة المفاتيح.
1) اطبع هذه الأعداد على الشاشة.
2) احسب مجموع هذه الأعداد المدخلة. 3) احسب عدد هذه الأعداد.

```
# include <iostream.h>
Void main( )
{
   int i , X , S ;
   i=0 ; S = 0 ; Cin >> X ;
  while( X!=0 )
  {   cout << X ;
      i = i+1 ;        ← عداد
      S = S + X ;      ← مجموع
      Cin >> X ;
  }
 Cout << "\n S =" << S ;
 Cout << "\n i =" << i ;   ← طباعة العدد هذه الأعداد
}
```

ملاحظة: للخروج من الحلقة نضغط على الرقم 0

* حلقة do-while (repeat-untile):

وصيغتها:

```
do
{  st1 ;
   st2 ;
   :
   st10 ;
}
while( exp ) ;
```

في حلقة while ومن حلقة do-while

في حلقة while يتم اختبار الشرط في البداية، وعند عدم تحقق الشرط الأمر لا ينفذ

AL AMANY Notebooks

Subject .... Date ....

أما في حلقة (do - while) يتم اختبار شرط نهاية الحلقة لذلك حلقة (do - while) تنفذ مرة واحدة على الأقل.

تمرين: أكتب برنامج لحساب مجموع الأعداد الصحيحة من 1 إلى 10.

```
i=0 ; n=10 ; s=0 ;
```

```
while (i<=n)
{ i=i+1 ;
  s=s+i ;
}
cout<<s;
```

```
do
{
  i= i+1 ;
  s= s+i ;
}
while(i<n) ;
      i<=n أو
cout<<s;
```

تمرين: أكتب برنامج يسمح بإدخال عدد صحيح من لوحة المفاتيح: (1) بين هل العدد زوجي أو فردي. (2) إطبع قواسم هذا العدد (3) احسب مجموع قواسم هذا العدد. (4) إطبع عدد القواسم.

Subject ⌋ Date ⌋

المتغيرات الحرفية Char : كل متغير حرفي يشغل حجم في الذاكرة مقداره (2 Byte)

وصيغته : اسم المتغير Char ;

Char C ;

C هو عبارة عن متغير حرفي .

عند إعطاء قيمة للمتغير الحرفي يجب وضع القيمة بين إشارتي تنصيص مفردتين : C = 'A' ;

من أجل إدخال المتغير الحرفي نستخدم : cin >> C ;

" " طباعة " " " : cout << C ;

ملاحظة : لكل متغير حرفي رقم ترتيب كما هو معمول في جدول (الأسكي كود) ، أي عند إعطاء الرمز يمكن معرفة رقم الترتيب ، وعند إعطاء رقم الترتيب يمكن معرفة الرمز .

مثال :

```
Char ch ;
   ch = 'A' ;
   Cout << ch ;  ——→  A
```

سوف يطبع A

مثال :

```
Char C ;
   C = 65 ;
   Cout << C ;  ——→  A
```

سوف يطبع الرمز المقابل A

مثال :

```
int C ;
   C = 'A' ;
   Cout << C ;  ——→  65
```

تمرين : اطبع جدول (الأسكي كود) الرموز ثم رقم الترتيب .

تمرين : اكتب برنامج لطبع الحروف الكبيرة .

- المصفوفات: إذا كان لدينا مجموعة من المتغيرات يمكن التعامل مع هذه المتغيرات تحت اسم واحد، إذا كانت المتغيرات جميعاً من نفس النوع فالقول لدينا مصفوفة.
هناك مصفوفات ذات البعد الواحد، ومصفوفة ذات البعدين.

- المصفوفة ذات البعد الواحد:
صيغتها: نوع المتغيرات اسم المصفوفة [عدد العناصر] ;
مثال:

```
int A[5];
```

هنا لدينا مصفوفة تدعى A مكونة من 5 عناصر، وجميع العناصر أعداد صحيحة. العنصر الأول بالمصفوفة يبدأ دليله بالرقم صفر، والعنصر الأخير ينقص دليله بمقدار واحد عن رقم المصرح عنه. أي:

A[0] , A[1] , A[2] , A[3] , A[4]

لإدخال عناصر مصفوفة ذات بعد واحد نحتاج إلى حلقة (For) واحدة. وكذلك بالنسبة للأمر الطباعة.

- عناصر المصفوفة السابقة يتم حجز مواقع لها في الذاكرة بشكل متجاور.

A[0]	A[1]	A[2]	A[3]	A[4]

عند تعريف المصفوفة لابد من تحديد عدد عناصر المصفوفة:
ويتم التحديد بالأمر:

```
#define n 5
const int n = 5;
```
أو

- م.ل: أكتب برنامج ضرب مصفوفتين ذات بعد واحد.

```
#include <iostream.h>
#define n 5
void main ( )
{
  int A[n], B[n], i, s;
```

Subject Date

```
for(i=0 ; i<n ; ++i)
    cin>> A[i] ;
for(i=0 ; i<n ; ++i)
    cin>> B[i] ;
S=0 ;
for(i=0 ; i<n ; ++i)
    S= S + A[i] * B[i];
Cout<<"\nS:"<<S ;
}
```

(نهاية الأمر ونهاية حلقة for)

- المصفوفة ذات البعدين وصيغتها:

نوع المتغيرات ← اسم المصفوفة [دليل أول] [دليل ثاني] ;

- مثال:

```
int A[3][4] ;
```

(3 ← عدد الأسطر، 4 ← عدد الأعمدة)

هنا لدينا مصفوفة اسمها A مكونة من ثلاثة أسطر وأربعة أعمدة.
العنصر الأول رقم الدليل يساوي الصفر والعنصر الأخير ينقص ~~منه~~ دليله بمقدار واحد.

A[0][0]	A[0][1]	A[0][2]	A[0][3]
A[1][0]	A[1][1]	A[1][2]	A[1][3]
A[2][0]	A[2][1]	A[2][2]	A[2][3]

- لإدخال عناصر المصفوفة ذات البعدين نحتاج الى حلقتين (for) واحدة للأسطر وواحدة للعمود.

- مثال: اكتب برنامج يسمح بإدخال وطباعة عناصر المصفوفة ذات البعدين.

```
# include <iostream.h>
# define   m   3
# define   n   2
Void   main(  )
{
  int A[m][n] ;
  for (i=0 ; i<n ; ++i)
    cin >> A[i] ;
```

AL AMANY Notebooks

Subject Date

```
for (i=0;i<n;++i)
  cin>> B[i];
S=0;
for(i=0;i<n;++i)
  S=S+A[i]*B[i];
  cout<<"\n S="<<S;
  }
```

البرنامج

```
#include <iostream.h>
# define m 3
# define n 2
void main( )
{
  int A[m][n],i,j;
  for(i=0 ; i<m,++i)
   {
     for(j=0;j<n;++j)
     {
       cout<<"\n A["<<i<<"]["<<j<<"]=";
       cin >> A[i][j];
     }
   }
  for(i=0 ; i<m ;++i)
   {
     for(j=0;j<n;++j)
     {
       cout<< A[i][j];
     }
```

Subject .... Date ....

- أكتب برنامج يقوم بما يلي : [1] - ادخال عناصر مصفوفة ذات بعدين .
  [2] - طباعة هذه العناصر .
  [3] - طباعة منقول مصفوفة
  [4] - طباعة عناصر القطر الرئيسي ثم اطبع عدد هذه العناصر .
  [5] - إطبع عناصر القطر الثاني .
  [6] - إطبع عناصر العمود الثالث .

```
#include <iostream.h>
#define n 5
void main ( )                         // مراد طباعة عدد عناصر القطر الرئيسي
int A[n][n], i, j, k, p;              // عناصر القطر الثاني
for(i=0; i<n; ++i)
  for(j=0; j<n; ++j)
    cin>> A[i][j];
    ~~cout<< A[i][j];~~
    for(i=0; i<n; ++i)         }
    for(j=0; j<n; ++j)         }  طباعة العناصر
     cout<< A[i][j];           }
    for(i=0; i<n; ++i)         }
    for(j=0; j<n; ++j)         }  طباعة المنقول.
     cout<< A[j][i];           }
    k=0;
    for(k=0, i<n, ++i)
     for(j=0, j<n, ++j)
      if( i==j)
      {
          k=k+1;  ——— عدد عناصر القطر الرئيسي
          cout<< A[i][j];  ——— طباعة العناصر
      }
```

AL AMANY Notebooks

Subject Date

```
cout<<"\n k="<<k;          ← طباعة عدد العناصر.
~~if(i+j==n+1)~~
  p=0;
for(i=0; i<n; ++i)
 for(j=0; j<n; ++j)
  if(i+j==n+1)
   {
     p=p+1;
     cout<< A[i][j];
   }
  cout<<"\n p="<<p;        طباعة عناصر القطر الثانوي
for(i=0; i<n; ++i)
  cout<< A[0][i];          → أيضا السطر الأول
         (صف) (عمود)
  cout<< A[i][0];          → العمود الأول
```

١- احسب مع الطباعة مجموع عناصر العمود الثالث للعمود الثاني ثم اطبع مجموع عناصر السطر الأول للسطر الأخير.

Subject | Date |

- المتغير Char :

```
Char c;
c = 'A';
Cout<< c; Cin>>c;
```

- التعامل مع السلاسل الحرفية :

أما باسكال

```
Var      c := 'A';
c :char ;
s : string
s1 : string(10) ;
  s := 'Basic';
```

- ذكرنا في المحاضرة السابقة أن المتغيرات الحرفية تعطى بالصيغة • char c
كل متغير حرفي يشغل حجم في الذاكرة مقداره (1Byte) وعند إعطاء قيمة للمتغير الحرفي
يجب حصر هذه القيمة بين إشارتي تنصيص مفردتين.

مثال

```
char c;
c = 'A';
```

في الواقع نوع المتغيرات char يمكن أن نستخدم أيضاً في التعامل مع السلاسل الحرفية
- السلسلة الحرفية : هي عبارة عن سلسلة أو تتابع خطي من الأحرف والأرقام
والرموز.
عند تعريف السلسلة الحرفية يجب أن نحدد طول السلسلة الحرفية أو عدد أحرف هذه
السلسلة :
وصيغتها :

char اسم السلسلة الحرفية [عدد الأحرف] ;

مثال :

char S[10] ; ⟵ سلسلة حرفية طولها الأعظمي 10 أحرف

- السلسلة في الواقع هي عبارة عن مصفوفة حرفية.
عند الإعطاء قيمة للسلسلة الحرفية يجب حصر هذه القيمة بين إشارتي تنصيص مزدوجة
" "
مثل ...
يمكن أيضاً لتعريف السلسلة الحرفية وإعطاءها قيم ابتدائية ،

Subject Date

اسم الطالب

مثال:
```
Char fn [8]
cin >> fn;
```

مثال:
```
Char A[6] = "maen"
A[0] = 'm';
A[1] = 'a';
A[2] = 'e';
A[3] = 'n';
A[4] = '\0';
```
→ الحرف الصفري

- الحرف الصفري: يعبر عن نهاية السلسلة الحرفية
عند تعريف السلسلة الحرفية يجب أن نحجز موضع في الذاكرة للحرف الصفري، والمترجم اللغة يضيف الحرف الصفري في نهاية السلسلة ولا يظهر في الطباعة.

- من أجل طباعة السلسلة الحرفية:
```
cout << A;
```
- ومن أجل إدخال عناصر المصفوفة (السلسلة الحرفية) نستخدم الأمر cin والصيغة:

مثال:
```
Char A[10];
cin >> A;
```
```
cin >> A;
```
- هنا يتم إدخال السلسلة الحرفية من لوحة المفاتيح ~~عند إدخال السلسلة الحرفية~~ من لوحة المفاتيح لإدخال السلسلة من دون وجود فراغ لأن الفراغ يعبر عن حل بس السلسلة.
- من أجل السلسلة الحرفية من لوحة المفاتيح يمكن أن نستخدم أمر:
```
gets ( اسم السلسلة ) ;
```
وهنا يسمح بوجود فراغ
- عند استخدام هذا الأمر يجب أن نستخدم الملف الرأسي:
```
# include <stdio.h>
```
مثال:
```
Char fn(20);
gets(fn);
```

Subject | .... Date | ....

تمرين 2 أكتب برنامج يسمح بإدخال سلسلة حرفية من لوحة المفاتيح :

Data Base first one

(1) اطبع هذه السلسلة . (2) اطبع هذه السلسلة حرف حرف .

(3) اطبع هذه السلسلة من النهاية الى البداية (4) اطبع الكلمة الأولى من السلسلة .

AL AMANY Notebooks